UNIVERSITY OF TARTU
Institute of Computer Science
Computer Science Curriculum

Heidi Korp

Agent Behavior Modeling in Roundabout Traffic

Master's Thesis (30 ECTS)

Supervisor(s):
Naveed Muhammad, Yar Muhammad

Tartu 2022

Agent Behavior Modeling in Roundabout Traffic

Abstract:

Autonomous vehicle (AV) industry has grown immensely in the last few years. Different aspects of assisted and autonomous driving, including perception, state estimation, motion planning etc. have received a lot of attention from the research and industrial community. Achievements in hardware industry have enabled to make real-time analysis about the situation in traffic. One of the major challenges that the AV industry faces today is understanding and predicting the behavior and future states of road users. Modeling such behaviors is not a trivial task and depends on multiple factors including traffic rules, the geometrical shape of the road, number of traffic participants etc.

In this paper we propose two methods for predicting the future action of a vehicle that is about to enter the roundabout. The first method is based on the Recurrent Neural Network (RNN) architecture and aims to predict the destination of a vehicle. The second method uses the information about Surrounding Vehicles (SV) in addition to the Target Vehicle's (TV) data to predict the course of action in terms of velocity. The results indicate that a correct assumption about the vehicle's destination can be achieved in less than 0.4 seconds and that taking the SVs' data into consideration is very helpful in modeling the vehicle's future behavior.

Keywords:

Autonomous driving, intelligent transportation systems, machine learning

CERCS: P170 Computer Science, numerical analysis, systems, control

Agentide käitumise modelleerimine ringristmikel

Lühikokkuvõte:

Autonoomse juhtimise valdkond on viimastel aastatel palju arenenud. Mitmed selle valdkonna alamliigid, nagu näiteks objektide tuvastamine, oleku hindamine, liikumise planeerimine, on pälvinud teadlaste ja ettevõtete tähelepanu. Edusammud riistvara valdkonnas on võimaldanud analüüsida liiklust reaalajas. Üheks suurimaks katsumuseks autonoomse juhtimise valdkonnas on liiklejate käitumisest aru saamine ja vastavalt nende tuleviku käitumise ennustamine. Nimetatud käitumise modelleerimine pole triviaalne ülesanne ning sõltub mitmest faktorist, näiteks liiklusreeglitest, liiklusraja geomeetrilisest kujust, liiklejate arvust jm.

Selles teadustöös on esitatud kaks meetodit ennustamaks tuleviku käitumist sõidukile, mis valmistub ringristmikule sõitma. Esimene meetod on loodud Rekurrentse Närvivõrgu arhitektuuri baasil ning selle eesmärgiks on sõiduki mahasõidu ennustamine. Teine meetod kasutab lisaks sihtsõiduki (*Target Vehicle*) andmetele ka lähedal asuvate (*Surrounding Vehicles*) sõidukite andmeid. Eksperimentide tulemustest selgus, et korrektne ennustus sõiduki mahasõidu kohta on saavutatav vähem kui 0.4 sekundiga ning et lähedal asuvate sõidukite andmete kasutamine on väga kasulik sõiduki käitumise modelleerimiseks.

Võtmesõnad:

Isejuhtivad sõidukid, intelligentsed transpordisüsteemid, masinõpe

CERCS: P170 Arvutiteadus, arvutusmeetodid, süsteemid, juhtimine (automaatjuhtimisteooria)

Table of Contents

1 Introduction

In recent years, automating different processes in robotics industry has advanced very quickly. One of such industries is the automotive industry where the end goal is to minimize the human's role as a driver. Although the traffic rules set some boundaries to the vehicles' behavior in traffic, there are still many operations that are easy for a skilled human driver, but very complex for an algorithm to solve [1]. In this research we focus on roundabouts – a type of intersection that does not have traffic lights. Roundabouts play an important role in modern traffic and studies have shown that roundabouts reduce traffic accidents (compared to intersections regulated by traffic lights), can reduce delays and improve traffic flow [2]. From an autonomous vehicle's perspective, estimating the target vehicle's (and also surrounding vehicles') future state in a roundabout is not trivial, but is essential to ensure that entering the roundabout would be safe and efficient [3].

1.1 Context

A roundabout is a type of intersection that is not regulated by traffic lights. One of the most important decisions a driver must make regrading roundabouts, is whether to stop, slow down or speed up right before entering the roundabout. Without traffic lights ordering the vehicles, the drivers must rely a lot on non-verbal communication and make accurate decisions based on the nature of the intersection, the distance and speed of other vehicles already traversing the roundabout and the driving style of other drivers in the area. It is common that drivers display behavior that does not exactly follow traffic rules. Some examples of such behavior include speeding and not giving way, slowing down when not necessary or derailing from the traffic lane. For a vehicle to be truly autonomous, it is important to study these kinds of behaviors and take them into consideration when deciding on the most suitable course of action. Such task is trivial for a human but can be very challenging for an autonomous vehicle.

For safe and efficient operation on roads, an autonomous vehicle should not only understand the current state of near-by road-users, but also proactively anticipate their future behavior. One part of the problem is to predict the behavior of pedestrians, and another includes predicting the behavior of other surrounding vehicles. Unlike pedestrians, the behavior of vehicles is constrained by traffic rules, the geometry of the roundabout and their high inertia that reduces the complexity of the task. However, one of the biggest challenges regarding the behavior prediction of vehicles arises from interdependency, where the actions of one vehicle affect the surrounding vehicles. Therefore, when predicting the behavior of one vehicle, it is necessary of observe the behavior of surrounding vehicles to make an accurate prediction.

The aim of this thesis is to propose the most probable course of action regarding speed for the vehicle that is about to enter the roundabout. The data used in this thesis is collected in Australia by the University of Sydney and Australian Center for Field Robotics [4].

1.2 Contribution

We introduce two methods for proposing a course of action for a vehicle about to enter a roundabout (target vehicle). The aim of the first method is predicting the vehicle's destination. By knowing the destination of the vehicles currently traversing the roundabout, it is possible to use this knowledge for modeling the target vehicle's future behavior. The model is a recurrent neural network that uses an LSTM layer with peephole connections. We compare three different network lengths: 5, 10 and 15.

The second method includes categorizing the data samples into groups based on their velocity: speeding up, slowing down, and waiting. The goal of such method is to help the target vehicle choose a suitable course of action when approaching a roundabout. Based on the surrounding vehicle's position and velocity we predict the course of action using Support Vector Machine model.

We present an updated version of the Five Roundabouts dataset [4] where every data sample is categorized into three groups based on their velocity: speeding up, slowing down or waiting. For this task, we have unnormalized the trajectories in the dataset as it is necessary to consider the actual location of a vehicle as well as the distance between the vehicles.

1.3 Structure

The reminder of the manuscript is structured as follows:

- Chapter 3 gives an overview of related work;
- Chapter 4 provides an overview of the technical background
- Chapter 5 gives an overview of the dataset
- Chapter 6 introduces the methodology of the research
- Chapter 7 lists the experiments done in this research and results
- Chapter 8 contains analysis and discussion about the results
- Chapter 9 introduces our suggestions of what to improve in the future

- Chapter 10 contains the conclusion of the research

2 List of abbreviations and terms

ANN – artificial neural network

BPTT – backpropagation through time

CEC – constant error carousel

Data point – one snapshot of a vehicle that includes the timestep, object identification, vehicle relative location and velocity, csv file name etc.

DNN – deep neural network

Ego Vehicle (EV) - autonomous vehicle which observes the surrounding environment to predict the behavior of TVs

LiDAR – Light Detection and Ranging

LSTM – long-short term memory

RNN – recurrent neural network

Surrounding Vehicles (SVs) – vehicles that are traversing the roundabout while the TV prepares to enter the roundabout

SVM – support vector machine

Target Vehicles (TVs) - vehicles whose behavior we are interested in predicting.

Track – a collection of data points. One track describes the full path of a vehicle from before entering the roundabout, its data during coursing about the roundabout and its information after exiting the roundabout

3 Related works

Behavior prediction is an important component in the software unit of an autonomous vehicle. Although there are numerous studies related to this topic, there are many scenarios that must be taken into consideration. The following sections will give an overview of different approaches to solving this problem.

3.1 Models for predicting agent behavior at roundabouts

Zyner *et al.* are the authors of the Five Roundabouts dataset [4], the same dataset that is used in this research. In 2018 they published research focusing on the prediction of a vehicle's destination. This method is based on recurrent neural networks [5] that takes data from LiDAR-based tracking system that is similar to the ones used in autonomous vehicles. Another study [6] conducted in 2020 has the same goal – to predict driver intent at urban unsignalized intersection, but in this experiment, it is achieved by using a multi-modal trajectory prediction algorithm with uncertainty and by clustering the result into a meaningful output. This method is also based on recurrent neural networks, but in addition they use a mixture density network output layer. The produced output is a multi-modal distribution over the predicted path a driver may take. Muhammad *et al*. [7] have analyzed two methods for agent intention estimation: particle filter algorithm; and decision trees. The experiments were conducted using three datasets that consist of real-world bicycle and car trajectories in two different scenarios – at a roundabout and a T-junction with a pedestrian crossing. The research [7] showed that the decision-tree based intention estimation works better on a binary-class problem, whereas the particle-filter based method works better on a multi-class problem, such as a roundabout.

Lefevre *et al.* [8] have conducted a survey in 2014 of the best methods at the time for motion prediction and risk assessment for intelligent vehicles. They have classified the behavior prediction models into three categories: physics-based, manoeuvre-based, and interaction-aware models. The simplest model is the physics-based model that only takes into account the laws of physics. The manoeuvre-based models predict the vehicle's behavior based on their intended manoeuvre. The interaction-aware model type is the most complex one, because in addition to other factors, it considers the interaction between vehicles. This paper was written in 2014, and at the time deep learning methods were not so popular as they are now for solving similar challenges. Physics-based models are not state-of-the-art anymore and thus, the authors of the article [8] do not include it in their classification. Instead, they classify existing work by three different criteria: input representation, output type, prediction method.

Mozaffari *et al.* published a review [9] in 2020 about the state-of-the-art deep learning-based approaches for vehicle behavior prediction. The grouping of input representation, output type, prediction method and their corresponding model types are described in Figure 1.

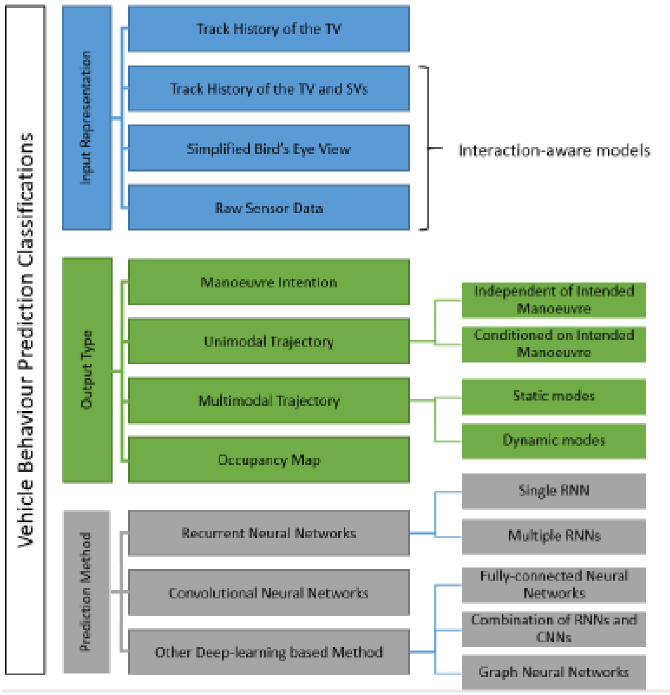

Figure 1. Proposed classifications of state-of-the-art deep learning approaches for vehicle behavior prediction. Replicated from [9]

In recent years, deep learning-based approaches have become popular due to their promising performance in more complex environments compared to the conventional approaches. However, there are practical limitations of implementing recent solutions in autonomous vehicles such as sensor impairments and limited computational resources, which have been described in [9].

3.2 Best features for behavior modeling

Some researchers [5], [10], [6] have used the track history of x-y position, speed and heading of the TV to predict its behavior at roundabouts. For instance, in research [7] it was found that the best features in exit-prediction classification task are heading angle and lateral position offset, while speed was the least discriminative feature. Furthermore, some studies have discussed categorizing vehicles into classes based on the vehicle type. In [7] it was found that for some categories the prediction of future position improves, while for some categories, it gets worse. In general, the authors have suggested that if the data about different vehicle categories is available, taking this into consideration can affect, and in some cases enhance the state prediction of agents in traffic.

However, in all of these studies, it is assumed that there are no Surrounding Vehicles (SV) in the environment. Few deep learning-based experiments have used the data from SV as input data [11], [12]. Xin *et al.* [11] argue that the information of SVs is not available due to Ego Vehicle's (EV) sensor limitations and occlusion problems. Nonetheless, excluding the information about SVs state from the input set may result in inaccurate prediction results because of the interdependency of the vehicles' behavior. [9]

3.3 Task and model grouping in roundabouts

Muhammad *et al.* [3] investigate the tactical task of entering a roundabout from an autonomous vehicle's perspective. The *tactical* in this context refers to the naming convention in studies [13], [14] where:

- *strategic* tasks comprise of high-level and long-term planning decisions such as route choice, cost estimates etc.
- *operational* tasks include low-level, short-term and continuous route tasks such as lateral control based on immediate environmental output [3]
- *tactical* refers to tasks that fall between those two categories, including, but not limited to turning, overtaking, gap adjustment etc. [13]

In their research, Muhammad *et al.* [3] have proposed a method for estimating the future position of an agent. The solution is based on three model types:

- geometric model of a roundabout
- data-driven mean model of a roundabout
- a model based on a set of reference trajectories traversed by agents at the roundabout

The geometric model is the most generalizable from the three and it is generated by grouping roundabouts based on their size, number of lanes and other geometric characteristics. The second model is got by using a set of actual vehicle trajectories for every path possibly available. This method is not as generalizable as it depends on the specifics of the roundabout the data was recorded at. The third solution also uses the vehicle trajectories recorded in naturalistic environment, but opposed to the second model, it uses the calculated average paths of the recorded trajectories as reference.

The experiments in [3] showed that the set of trajectories model performs the best regarding both convergence time and state prediction in terms of mean absolute Euclidean distance error between predicted and ground-truth positions (1, 2 and 3 seconds into the future). Since this model is the most specific one (and the geometric is the most generic one), it could be expected that the model based on set of trajectories performs the best.

3.4 Driving styles

In addition to aforementioned factors such as the geography of the roundabout, number of road participants and traffic rules, there are some other factors that can affect the agents' driving style. Ericsson [15] has conducted a research comparing the relationship between external conditions such as traffic, street type and other drivers. Besides these factors, it was found that the time of the day and the day of the week may affect drivers differently. The most important environmental factors that contribute to the driving style were found to be traffic conditions and road type [15]. Some researchers [16], [17] have taken a different approach by analyzing human factors such as conscious decision making and demographic background. The results showed that multiple internal factors, such as fatigue, driving experience, familiarity with the vehicle and the environment may strongly condition the driving style. Furthermore, the same driver could exhibit separate styles under different conditions such as driving to work during rush hours or going on a family trip during the weekend.

Martinez *et al*. [18] have conducted a survey of different ways for driving style recognition. Depending on the quantity and variability of the information available, the driving style recognition methods can be grouped into three categories: the models implemented through rules; model-based algorithms and models using machine learning techniques. These categories have been described in Figure 2.

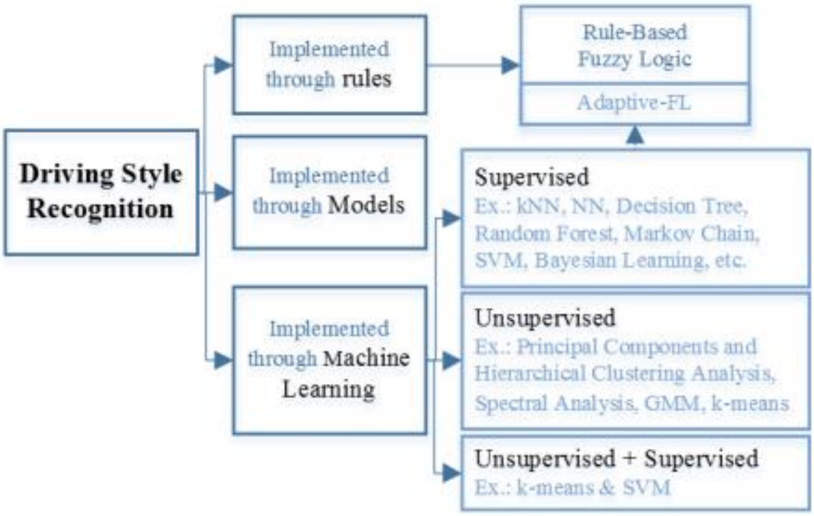

Figure 2. Summary of driving style recognition algorithms. Replicated from [19]

Safety related systems would benefit from knowledge about driving styles to predict and anticipate the drivers' reactions and to adjust to individual users. Driving style recognition may enhance the collision avoidance systems as well as help make vehicles more eco-friendly by managing the fuel consumption based on the driving style [19]. Furthermore, it has been argued that when drivers know that their driving is being monitored, then this motivates them to correct their driving style [20]. Better understanding of driving style is required to ensure appropriate and consistent recognition and to efficiently promote safety and eco-driving.

4 Technical background

This chapter provides an overview of the technical terms and methods used in this research.

4.1 Neural Networks

Neural networks, also known as artificial neural networks (ANNs), are a subset of machine learning and are the main component in a deep learning algorithm [21]. The idea for the algorithm originates from the study of the human brain where neurons exchange information by passing it to one another. The goal of this algorithm is to recognize the relationships between vast amounts of data. Neural networks are used in many different fields of life including autonomous driving, robotics, finances, medicine and more.

Every Artificial Neural Network (ANN) consists of at least three layers: the input layer, one or more hidden layers and an output layer. A network that has multiple hidden layers is called a deep neural network (DNN). In every layer, there are multiple nodes (or neurons) that are connected to one another. Every node is assigned a weight that determines the importance of that variable and threshold that specifies whether information will be passed to that node or not. Information will be passed to the input layer and if a neuron in that layer is above the threshold, the neuron will be activated, and the information will be passed to the next layer through that neuron. The same algorithm will continue until the final layer is reached and a prediction is made. This type of network is called a feed-forward neural network.

The previously described algorithm will be performed multiple times. The process where the whole data is passed through the network is called an epoch. After every epoch, the gradient of error will be calculated given the error function, with respect to the network's weights. The gradient of the final layer of weights will be calculated first and the gradient of the first layer of weights will be calculated last. This algorithm is called the backward propagation of errors or "backpropagation" for short. [22]

4.2 Recurrent Neural Network

A recurrent neural network is a type of deep neural network that uses sequential or time series data [23]. When a regular neural network assumes that the input of the previous layer is independent from the input of the current layer, then the recurrent neural network uses the past layers input to influence the current layer's input and output. Another difference is that opposed to a feedforward neural network that has different weights across each node, a recurrent neural network shares parameters across each layer of the network. The illustration of RNN and a Feedforward NN have been presented on Figure 3.

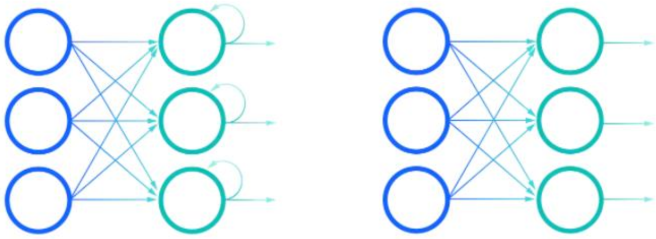

Figure 3. Difference between Recurrent Neural Network (left) and Feedforward Neural Network (right). Replicated from IBM Cloud Education [23]

Recurrent neural networks use the backpropagation through time algorithm (BPTT) that is slightly different from the traditional backpropagation. The main idea of the algorithm is the same: as the name suggests, the algorithm implements the normal backpropagation to RNN backwards through time.

One issue that is very common regarding recurrent neural networks, is called the vanishing/exploding gradient problem [24]. The problem lies in the fact that for long input-output sequences, RNNs have trouble modeling long-term dependencies. For long sequences, the model has to remember the historical representation of all the elements, meaning that the longer the sequence, the more parameters are in the algorithm. The problem arises from the chain rule in backpropagation algorithm that includes calculating products of partial derivatives. The number of factors in the product for early time slices is proportional to the length of the input-output sequence. Unless the products are all close to 1, two problems might occur: if the partial derivatives are less than 1, the product will become very small, called vanishing, or if the derivatives are more than 1, then it will result in the product blowing up, meaning that the product will become very large. This results in the learning to be very slow (in the vanishing case) or very unstable (in the exploding case).

Recurrent neural networks are used in many domains such as speech and text recognition, image processing and analysis of historic data that occurred in a specific sequence.

4.3 Long-short term memory

Long-short term memory (LSTM) is a popular RNN architecture introduced by Sepp Hochreiter and Jürgen Schmidhuber [25]. This algorithm helps overcome the vanishing/exploding gradient problem so that the RNN algorithms can safely be applied to very long sequences. Thanks to the LSTM algorithm, it has been possible to improve results in Natural Language Processing domains, such as speech and text recognition, language modeling and machine translation.

The basic unit of an LSTM network is named the *memory block* containing one or more *memory cells*. In the memory block, there are three adaptive, multiplicative gating units that is shared among all cells. The information is stored in every cell, where there is a recurrently self-connected linear unit called in some researches as the "Constant Error Carousel" (CEC) [26]. This unit is responsible for recirculating activation and error signals indefinitely, which allows short-term memory storage for extended time periods. The three gates can be learned, respectively, what information to store in memory, and when to read it out. By combining memory cells into blocks, the cells can share the same gates which reduces the number of adaptive parameters. The role of each gate will be described in the following paragraphs.

The LSTM unit consists of a cell, an input gate, an output gate and a forget gate. The forget gate is responsible for removing information that is no longer useful in the cell state. It takes two inputs – the current input and previous cell output and multiplies them with weight matrices and adds the bias. The result is passed through an activation function that gives a binary output. If the cell state output is 0, then this information is forgotten and similarly, if the output is 1, then this information is retained for future use.

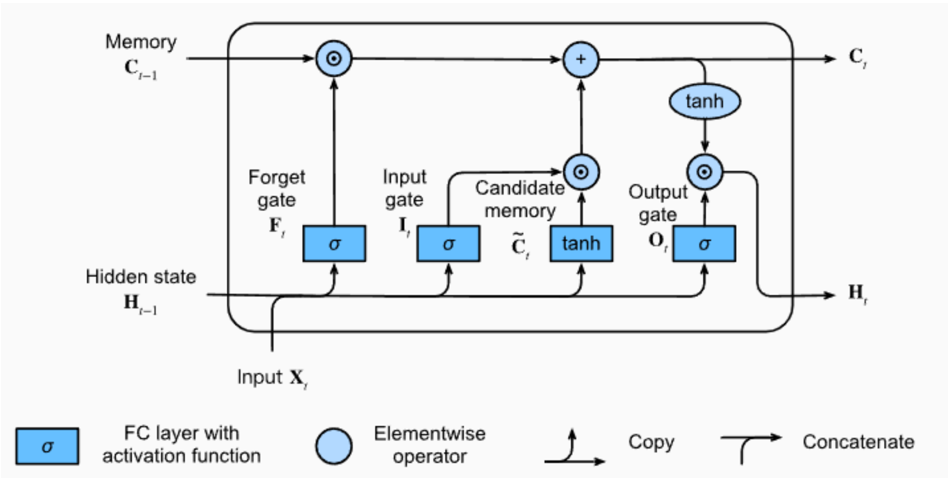

Figure 4. Illustration of the LSTM architecture. The forget, input and output gates are marked on the illustration along with application of the tanh and sigmoid functions. Figure is replicated from [27]

The input gate is responsible for adding useful information to the cell state. First, the information is regulated using the sigmoid function and the values are filtered as in the forget gate using the values of current input and previous cell output. Next, a vector is crated using the *tanh* function that produces an output ranging from -1 to 1 that contains all possible values from the previous cell output and current cell input. Finally, to obtain the useful information, the values of the vector and the regulated values are multiplied. The LSTM architecture is described in Figure 4.

The output gate is responsible for extracting useful information that will be passed on to the next cell. First, the *tanh* function is applied on the cell to get a vector. The information is then regulated using the sigmoid function and filtered. Finally, the values of the vector and the regulated values are multiplied to form the output of the current cell.

4.4 LSTM with Peephole connections

One of the limitations of traditional LSTM is that each gate receives connections from the input units and the outputs of all cells, but it has no direct connection with the CEC it is supposed to control. In a situation where the output cell is closed, the cell output remains close to zero and the cells have no access to the CECs they control. This prevents the network from accessing important information and as a result, the performance may decrease.

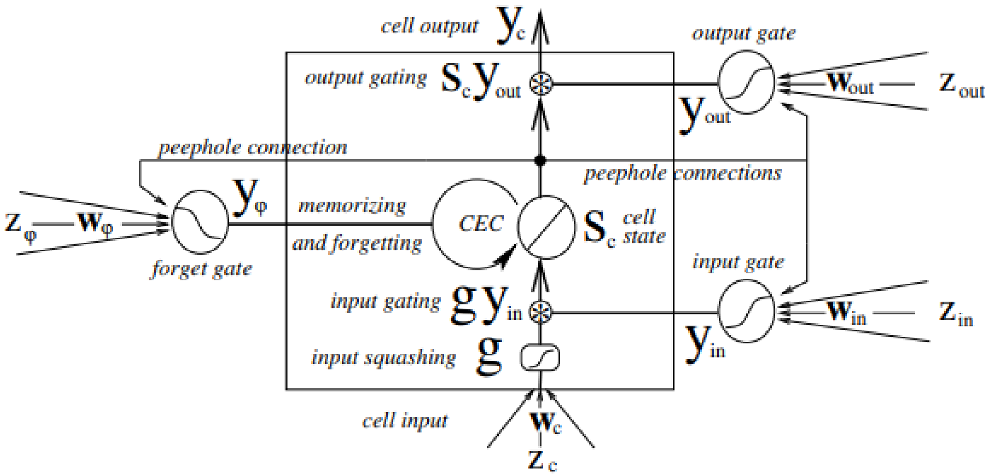

Figure 5. LSTM memory block with peephole connections from the CEC to the gates. Replicated from [26]

A simple solution to solve the problem is to add weighted "peephole" connections to every memory block from CEC to the gates (Figure 5). This connection enables all gates to inspect the current state even when the output gate is closed. Peephole connections are treated like regular connection to gates and no error signals are propagated back to the CEC during learning.

4.5 Support Vector Machine

Support Vector Machines (SVM) are a set of supervised learning methods used for classification, regression and outliers detection. Some of the advantages of this method is that it is effective in high dimensional spaces and in cases where the number of dimensions is greater than the number of samples. As it uses a subset of training points in the decision function that are called support vectors, it is also memory efficient. Kernel functions make the SVM a versatile method because different kernel functions can be specified depending on the dataset. In addition to common kernels, it is also possible to define custom kernels.

In order to avoid overfitting, it is crucial to use kernel functions and regularization if the number of features is much greater than the number of samples. Another disadvantage is that SVMs do not provide probability estimates and they have to be calculated using an expensive five-fold cross-validation.[28]

5 Dataset

In order to conduct a thorough analysis for the behavior of drivers in the roundabout, the chosen dataset has to be large enough to capture different situations that illustrate traffic in real world. A few datasets were considered and finally the ACFR Five Roundabouts [4] dataset was selected.

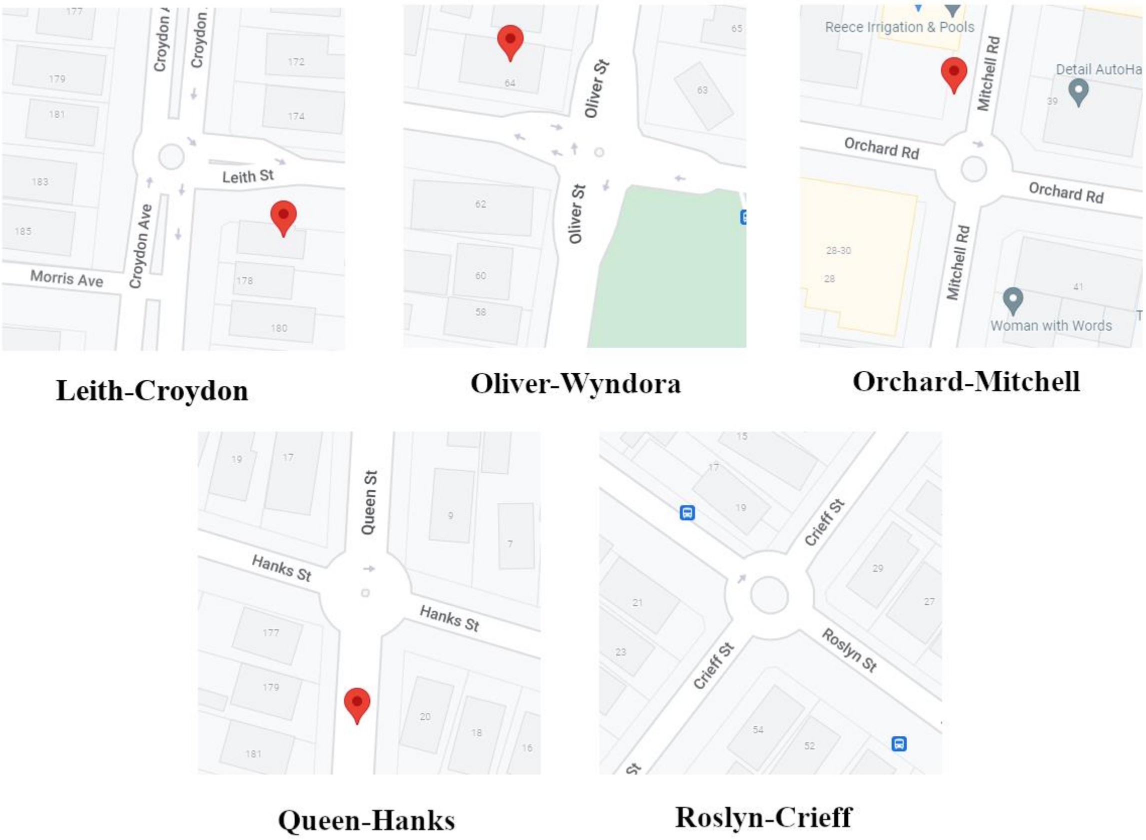

Figure 6. Illustration of the five roundabouts using the same scale

This dataset consists of 23,000 vehicle trajectories of real world naturalistic driving that makes it one of the largest datasets focusing on roundabouts. The aim for creating the Five Roundabouts dataset was to capture a high volume of naturalistic paths taken at single lane unsignalized intersections. The data is collected from five different roundabouts (Figure 6) in Australia, using a LiDAR-based detection and tracking system by Ibeo GmbH [29] on board a vehicle.

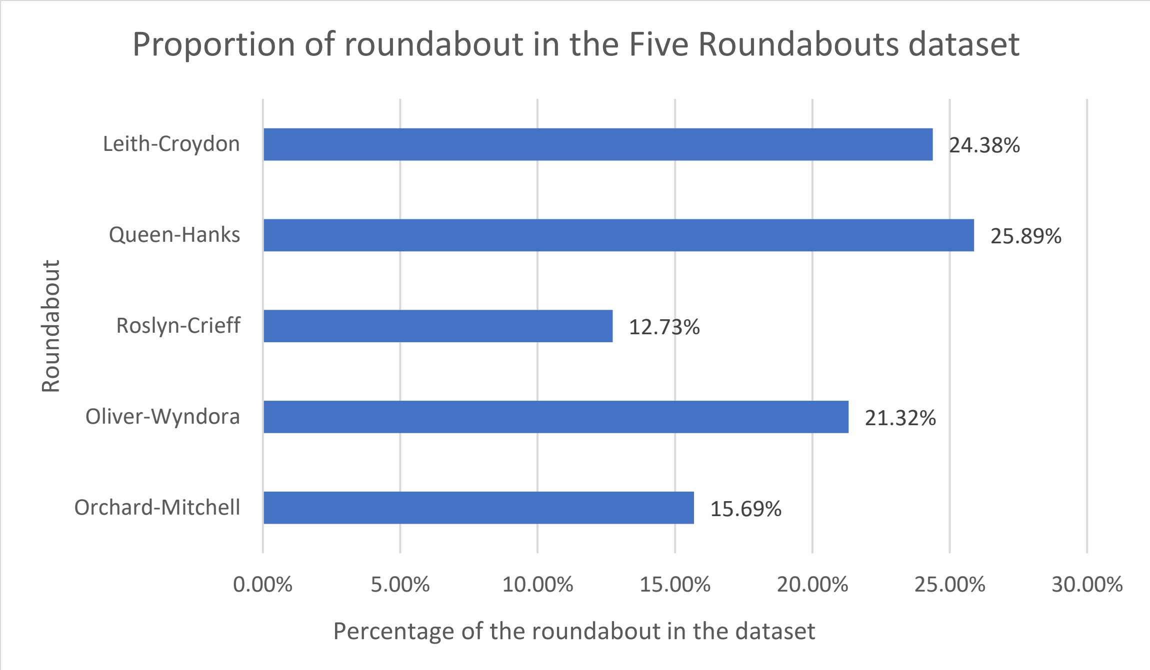

Figure 7. Proportion of each roundabout data in the whole dataset

Figure 7 describes the proportion of each roundabout in the whole dataset. The least data has been collected from the Roslyn-Crieff roundabout and this is also the only roundabout in the dataset where roundabout exits have been marked with “SW”, “SE”, “NW” and “NE” opposed to “North”, “South”, “West” and “East” as it for in the other roundabouts. When doing the analysis this information can be taken into consideration to determine, how well can a model generalize where there is less variety for some classes.

5.1 Conditions for dataset collection

This kind of data, such as the Five Roundabouts dataset [4], is mainly used for training and verification of maneuver-based models that predict which turn a vehicle will make at an intersection, and path prediction models that will predict the whole trajectory of the vehicle traversing an intersection. While gathering the data, the authors of the dataset took into consideration that a good dataset for training autonomous vehicles must capture the nuances in driver behavior such as negligent driving, speeding or assertiveness. This can only be achieved if the dataset is collected publicly where the drivers are not aware of the dataset being collected.

Another aspect that must be considered when creating the dataset is whether to use onboard sensors or overhead sensors installed at intersections. Since most of the intersections are not equipped with intelligent sensors, the vehicle must be able to make decisions based on information gathered by its onboard sensors. Furthermore, if the algorithm is used with onboard sensors, it is not guaranteed that the algorithm will perform similarly if it was trained with data gathered from overhead sensors.

5.2 Australian roundabout design

The geometry of a roundabout is proved to have great impact on the road users' driving style [15]. Since the Australian roundabouts have very different geometrical features compared to the European ones, the following section gives an overview of the main differences and challenges of the Australian roundabout type.

Australia practices left-hand traffic which means that a vehicle enters the roundabout from the left-hand side of the center island. Regarding the design, the main difference lies in the shape of the roundabout: the Australian roundabouts are tangential whereas the European ones are radial. The European roundabouts force the driver to slow down before entering the roundabout since the turns are sharp. The tangential roundabout does not reduce the speed of the driver much and enables a smoother traffic flow. In addition to encouraging faster travel, which equates higher throughput, the Australian roundabouts have good visibility. Figure 8 illustrates both of these roundabout types.

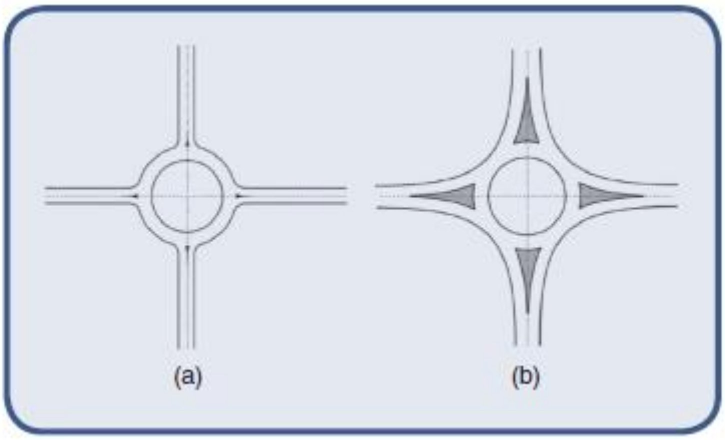

Figure 8. European style radial (left) and Australian tangential (right) roundabout. Reproduced from [30]

Another feature of these roundabouts is that they are small in size, with the center island being 10 meters in diameter, and they have good visibility. The rule for navigating in the roundabout is that the vehicle that enters the intersection first has right of way. However, due to having good visibility and the center island being small, assertive behavior is common even though it may be contrary to the road rules. Making the right decision to enter the roundabout safely requires experience and good understanding of the roundabout traits. To increase the model's ability to translate well to unseen data, the dataset is collection of data from five different roundabouts. The trajectory of two vehicles at an Australian roundabout have been illustrated in Figure 9.

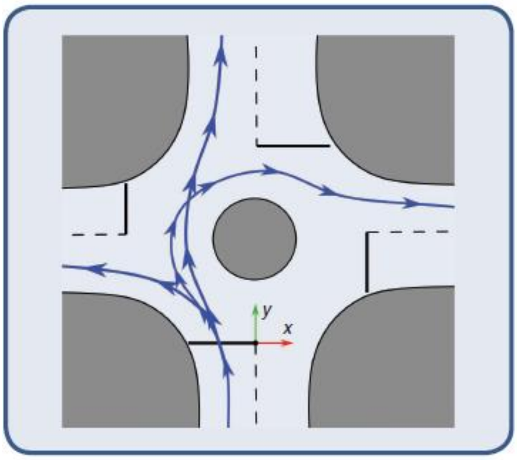

Figure 9. A diagram of a typical single lane roundabout, reproduced from [4]. On this diagram, all recordings are normalized so that the vehicle enters from south. The black line marks the entrance of the intersection. The coordinate frame is marked at the bottom of the diagram.

Even though the dataset is large, it does not cover the European style roundabouts, thus there remains the possibility that the models trained with this data are not well suited to use in Europe.

6 Methodology

This chapter describes two methods proposed in this thesis to predict the behavior of the target vehicle. In both cases the actual direction of entry and accurate position at each timestep are important in making the prediction. Considering this, a pre-processing step is needed to unnormalize the data described in Chapter 5. In the following sections we will give an overview of the problem at hand, pre-processing steps and the two methods that were proposed to solve the problem.

6.1 Description of the problem

The Five Roundabout dataset can mainly be used with two types of models: a model that predicts the vehicle's roundabout exit and a model that predicts the future trajectory of a vehicle. The goal of this research is to create a model to predict whether the TV can enter the roundabout right away or if it has to wait for a certain period of time before entering. Inspired by the nature of the problem, a new dataset was created based on the Five Roundabouts dataset. In addition to the original data, we have grouped the data points based on velocity into three states: waiting, slowing down and speeding up. The following paragraphs explain in detail, how was the dataset created and what kind of models were used to predict the vehicles' behavior.

6.2 Dataset pre-processing

All tracks in the Five Roundabouts dataset have been normalized in a way that the entrance to the roundabout is always in the south. For getting accurate results in this research, it is required to unnormalize the data to their original coordinates.

The Five Roundabout dataset has been pre-processed in a way that all tracks start from the south, at exactly 180 degrees. However, the actual roundabouts have multiple exits and not all the exists are at equal distances from each other. In order to accurately train the model, it is necessary to rotate the tracks to their original positions so that the direction of movement and distance would be correct. Figure 10 illustrates one of the five roundabouts of the dataset.

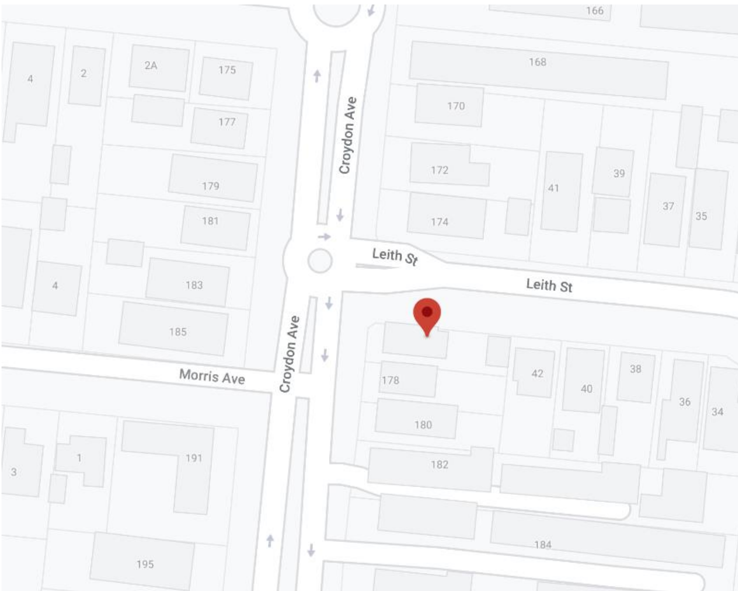

Figure 10. Illustration of the Leith-Croydon roundabout. The location of the recording (ego) vehicle is marked with the red marker. Snapshot reproduced from Google Maps.

Since the authors of the dataset have not provided detailed information about the pre-processing, we have estimated the center point of the roundabouts as well as the angle in which the tracks are rotated. The estimations were created by acquiring the aerial view of each roundabout from Google Maps application and they are illustrated in Figure 11.

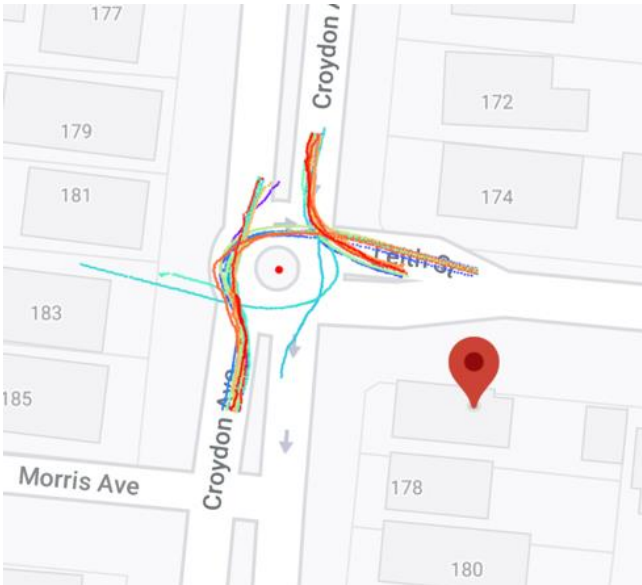

Figure 11. Illustration of a few tracks on the Leith-Croydon roundabout. The red dot at the center of the roundabout marks the center point about which the tracks have been rotated. The rotation angle was determined by fitting the tracks on the Google Maps image.

Then, the center point of the roundabout was determined by using trial and error method. By knowing the origin of the track and the center point, all the tracks were rotated by 90 angles at first and afterwards, each direction's rotation was further adjusted to match the roundabout shape seen in Google Maps.

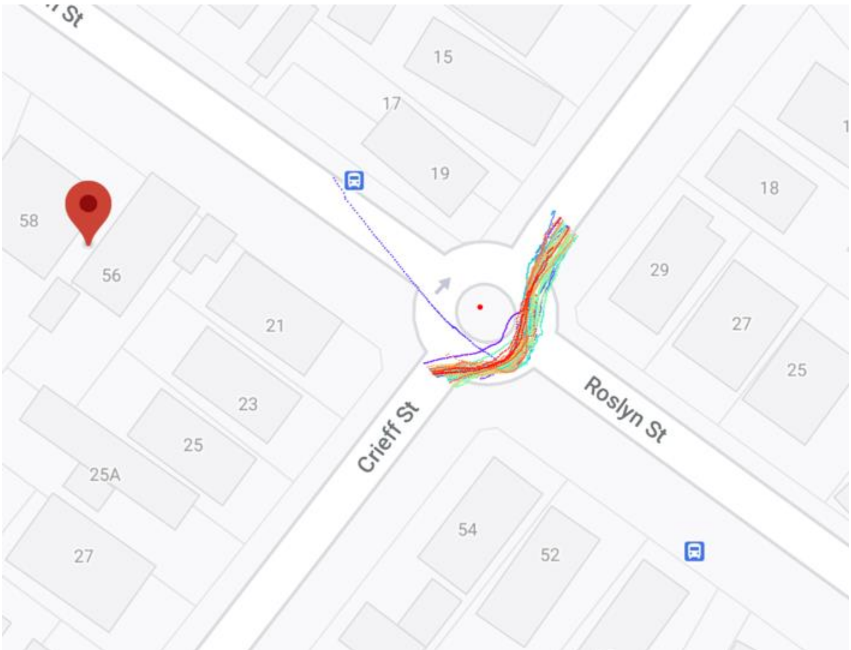

Figure 12. Illustration of the rotated tracks on the Roslyn-Crieff roundabout.

The Roslyn-Crieff (Figure 12) roundabout is the only intersection in the dataset where instead of north, south, east, and west directions the directions have been marked with south-east, south-west, north-east and north-west. The data collected from this roundabout makes around 13% of the whole dataset. Although the dataset is balanced across all possible exits, the results would be more accurate if there was the same number of examples per each exit.

6.3 Vehicle destination prediction

The first method that we propose to predict the behavior of a vehicle uses a model that predicts the vehicle's destination. The idea for the model's structure is inspired by Zyner *et al.* [5]. In this thesis we build our own model upon their work and the following sections will give an overview of the model's architecture and the functionality for deciding the driver's intention.

6.3.1 Model's architecture

The model's architecture used in this research is based on an RNN because these networks are especially useful for dealing with time series data. The model is used to interpret time series data about an externally observed vehicle. While training, there is one recurrence of the network per time-step, and after a chosen number of time-steps the prediction is made. The architecture of the RNN used in this model is shown on Figure 13.

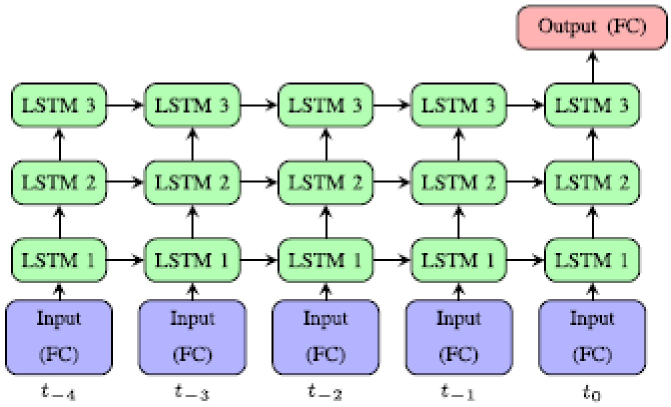

Figure 13. A diagram describing the architecture of the model proposed by Zyner et al. Here the model is input with a sequence of length 5, that ends at the nominated time-step t. the model consists of a single fully-connected layer, three LSTM layers and an output layer. Figure is reproduced from [5]

The LSTM layer used in the research is an LSTM with peephole connections (Chapter 4) that enables the model to access useful information that would be hidden with a classical LSTM layer. Instead of using three stacked LSTM layers, in the research at hand we used only one LSTM layer to create a moment of comparison and to speed up the computation time. In some studies [31], [32] it has been found that single-layer LSTM models have similar accuracy compared to multi-layer models, which was the motivation to use this technique in the current research.

To avoid losing important information, the dropout layer was not implemented in this research. With the normalized dataset, there were only three possible exits since south was declared the entrance of all tracks and U-turns were removed from the dataset as they were very rare. After denormalizing the dataset, the number of possible exits however, increased from 3 to 8 – one per each compass point.

The input features used in this model are X/Y position relative to the vehicle, heading in radians, and speed in meters per second. The data is normalized over the entire dataset, and then input into the network. A single dense layer was used as an input layer, of width 256 nodes and the following LSTM layer was 512 nodes wide. On the last layer, a softmax classifier is applied and the output of the network is one of the eight destinations: North, South, East, West, NE, SE, SW, NW. The overall goal of the model is to correctly predict the destination as early as possible.

ADADELTA [33] training was used, with the learning rate of 0.03. The dataset was split into 55% training, 20% validation and 25% for testing, with 4560 tracks for training, 1658 for validation and 2074 for testing, correspondingly. The network lengths used in this research were different than the ones used in the original research. Instead of lengths 5, 15, and 25 that correspond to 0.2, 0.6 and 1 seconds of data, respectively, we used lengths 5, 10, and 15 that correspond to 0.2, 0.4 and 0.6 seconds of data. The model was written in Tensorflow [34] and the model was trained on Nvidia GeForce RTX 2060 GPU.

6.3.2 Dataset pre-processing

Before training or evaluating time series data, it has to be preprocessed. Each track in the dataset is split into every possible sequential sequence that can be input into the RNN. The length of the sequence is determined by the number of steps in the RNN. For model evaluation the train/validation/test split is done track-wise to ensure that no training data overlaps the test or validation set. Seeding the random choice of trajectories for testing was kept fixed for the sake of consistency between experiments.

In order to properly score the network, the tracks are aligned by distance traveled from the entrance of the intersection. The first step to achieve this goal is to calculate the distance from the center point for each point in the track. Next, the data is grouped by roundabouts and by exits and the starting point is determined for each track. The starting point that is closest to the center point will be the threshold for this exit. Lastly, this threshold is set as the starting point for all tracks that originate from this exit. This helps to evaluate the destination prediction with different network lengths as all tracks start from the same position.

The metrics used to evaluate the model were accuracy and loss calculated using categorical cross entropy. The overall analysis was done by using the *evaluate* function from sklearn library [35] to compare the predicted values to ground truth. For the evaluation per exit, the predictions were done for every data point (a track consists of multiple data points) separately. If there were no correct predictions for data points at the end of a track, then this was considered to be an incorrect prediction. For the correct predictions, the last wrong prediction in a track was considered to calculate the convergence times.

6.3.3 Action suggestion based on SVs destination

After knowing the destination of the surrounding vehicles in the roundabout, it is possible to make a suggestion of what kind of action should the target vehicle take. After predicting the destination of a vehicle, its position is used to determine if it has already passed the target vehicle.

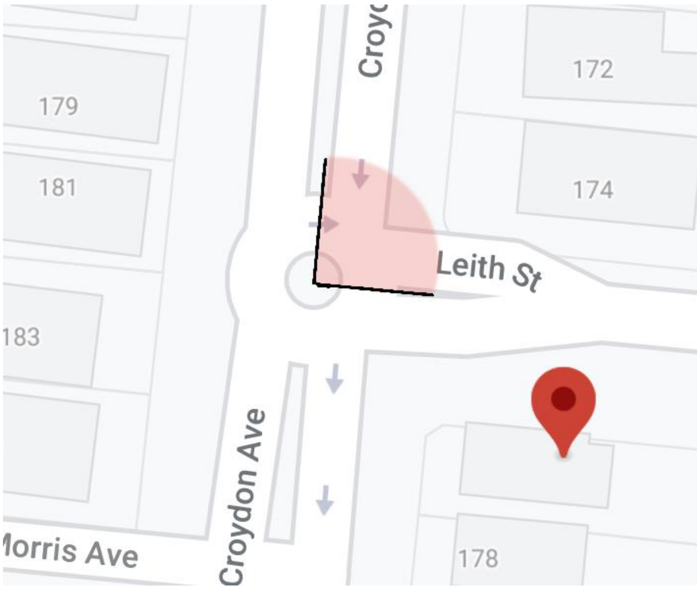

Figure 14. Illustration of the Leith-Croydon intersection. If a vehicle is in the area marked with red, then the TV has to wait

The vehicle is considered not to have passed if its position is in the right quarter from the TV and its destination. This area has been marked in red in Figure 14. This information is used to determine whether the TV should wait or not wait.

6.2 Model with SVs data

The second model we propose aims to predict the future action for an autonomous vehicle approaching a roundabout. The model uses SVs' data as input in addition to the TV's data. The following sections will give an overview of the steps in pre-processing and the architecture of the model.

6.2.1 Roundabout entry threshold

Since the decision of whether it is safe to enter the roundabout or not must be made before entering the roundabout, it is necessary to set a boundary before which the vehicle is considered not to have entered the roundabout yet, and after which the vehicle is considered to be traversing the roundabout. For this reason, we have defined a threshold that separates the normal road from the roundabout for each roundabout and each exit, which is illustrated on Figure 15.

Figure 15. An illustration of the Queen Hanks intersection. The location of the ego vehicle is marked with the red marker. The first 5000 data points are plotted on the figure, each track having a different color. The blue dots mark the beginning of each track and the black line defines the threshold that marks the entry of the roundabout. The red dot at the center marks the center point of the roundabout.

Using the roundabout illustration from Google Maps and by plotting the rotated tracks on the illustration, four points were determined that would help create the threshold for each exit. Next, a function was created that connects the four points, forming a tetragon. The angle of the threshold matches the angle of the roundabout exits.

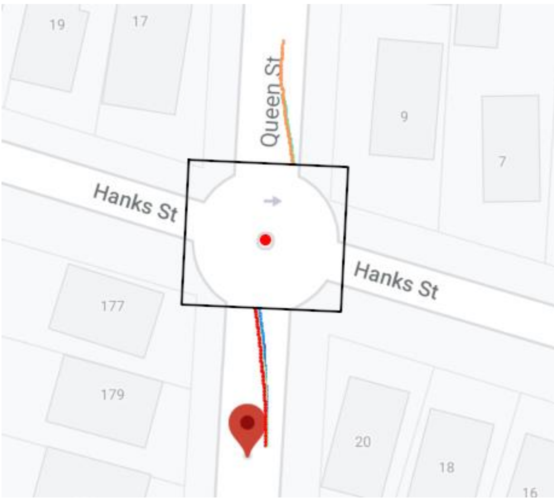

Figure 16. Illustration of the tracks before the threshold at Queen-Hanks roundabout.

Finally, a function is created that determines if a point is “inside” the threshold, meaning that it is closer to the center point, or if it is “outside” (Figure 16). Using these functions, a dataset containing only the track points that lay outside of the threshold is created.

6.2.2 Other vehicles in the roundabout

Due to the interdependency of vehicles, it is important to take into account the SVs data when predicting the most probable course of action for a vehicle. The number of vehicles, as well as their position and velocity may be one of the main reasons why a vehicle slows down or has to wait before entering the roundabout. Since we are looking only the part of each track that is before the entry threshold of the roundabout, we have created a separate dataset containing only that part of the track.

For each data entry, the whole dataset is searched for other entries where the timestamp difference is no more than one second. All such entries are called Surrounding Vehicles (SVs) and their relative position and velocity are added to the dataset. The SVs that are waiting too are removed from the SVs list.

6.2.3 Waiting, slowing and speeding vehicles

There are three actions that a target vehicle can take: wait, slow down or speed up. This is why the existing dataset is split into three categories based on the velocity. The vehicles whose relative velocity is zero, are labeled as waiting vehicles. For the rest of the vehicles, the average velocity is calculated and based on that, the dataset is split into two categories: slowing down vehicles where the velocity is below the average; and speeding vehicles where the velocity is above the average. A situation where multiple vehicles are at the same roundabout at the same time and where there are all three velocity categories present, is illustrated in Figure 17.

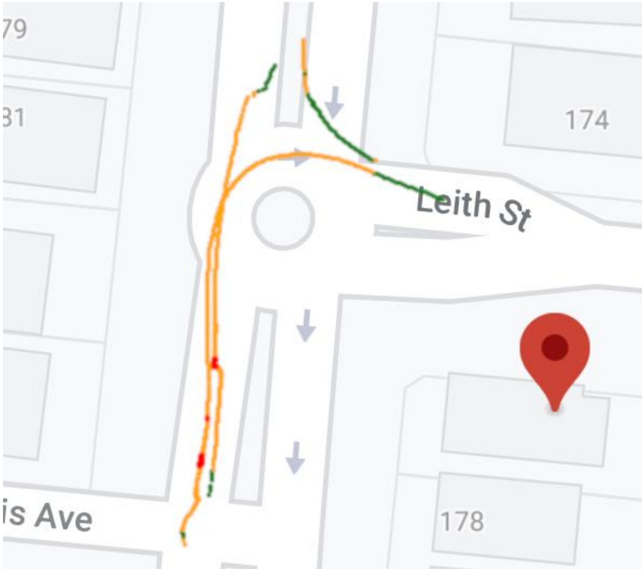

Figure 17. Illustration of three vehicles' trajectories. The datapoints where the vehicle is waiting, is marked with red. Yellow and green colors represent datapoints where the vehicle is slowing down and speeding up respectively.

By plotting the velocities, it was seen that there is no prominent threshold to separate speeding up and slowing down vehicles, so an average of the velocities was used. The Figure 18 illustrates the velocities across all data points.

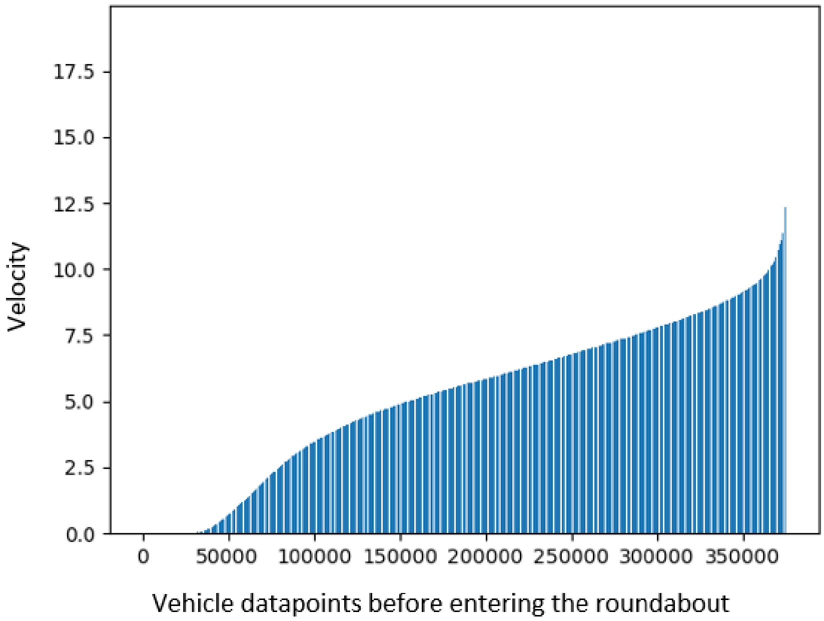

Figure 18. A diagram of all the datapoints' velocities that were recorded before entering the roundabout.

This method was further validated by calculating how many SVs there are at each timestep in each category. It was found that the most SVs were in the waiting category; the slowing down category had more and the least SVs were in the speeding category. It validates this splitting method because it is very probable that the vehicles that are waiting are doing that because there are many other vehicles in the roundabout.

The model was validated by calculating the accuracy using the *evaluate* function from sklearn library [35].

6.3 Dataset including information from SVs

Besides categorizing the TVs based on their velocity, it is important to know the information about other vehicles currently traversing the roundabout. In this experiment we used x- and y-coordinates as well as relative velocity of the surrounding vehicles in our training dataset. For every vehicle that had not yet entered the roundabout and for every timestep in its track, we matched that timestamp with any other vehicle's timestep. As a result, we gathered position and velocity data about all surrounding vehicles that were at the roundabout at the same time.

SV_X_1	SV_Y_1	SV_RelVelX_1	SV_RelVelY_1	SV_X_2	SV_Y_2	SV_RelVelX_2	SV_RelVelY_2	SV_X_3	SV_Y_3	SV_RelVelX_3	SV_RelVelY_3
3.692363	17.9126	-3.17	-0.64	5.300335	35.15942	-5.77	-0.38	4.006286	26.16776	0	0
3.692363	17.9126	-3.17	-0.64	5.300335	35.15942	-5.77	-0.38	3.959393	25.88067	0	0
3.692363	17.9126	-3.17	-0.64	5.300335	35.15942	-5.77	-0.38	3.878618	25.39397	0	0
3.692363	17.9126	-3.17	-0.64	5.300335	35.15942	-5.77	-0.38	3.851744	25.24266	0	0
5.300335	35.15942	-5.77	-0.38	3.692363	17.9126	-3.17	-0.64	3.626894	24.15382	0	0
5.300335	35.15942	-5.77	-0.38	3.692363	17.9126	-3.17	-0.64	3.512819	23.3219	0	0
5.300335	35.15942	-5.77	-0.38	3.692363	17.9126	-3.17	-0.64	3.494976	23.15042	0	0

Table 1. Section of the dataset including SVs. Here, there are two SVs traversing the roundabout at the same time as the TV, and dummy data has been used for the third SV.

By analyzing the dataset, we found that the maximum number of other vehicles at the roundabout at the same time, was 3. For every SV we added four new columns: *x_position*, *y_position*, *relative_velocity_x* and *relative_velocity_y*. In case there were less than three SVs or there were no SVs at all, we added some dummy data where the x- and y-coordinates matched with the TV's position and where velocity was zero. A sample of the created dataset is presented in Table 1.

Since the aim of the algorithm is to be able to predict the future action of a vehicle, all the action labels are shifted one data point upwards so that each data point is matched with the next data point's action label.

7 Results

In this chapter we have presented the results for experiments that focus on predicting the destination of a vehicle and a model that focuses on predicting the future action of a vehicle.

7.1 Destination prediction

We trained three models of different lengths to predict the destination of a vehicle. The model with network length of 5 had an accuracy of 12.2% while the model with network length 10 had an accuracy of 52.5%. The model with length 15 had an accuracy of 43%. The results are displayed in Figure 19.

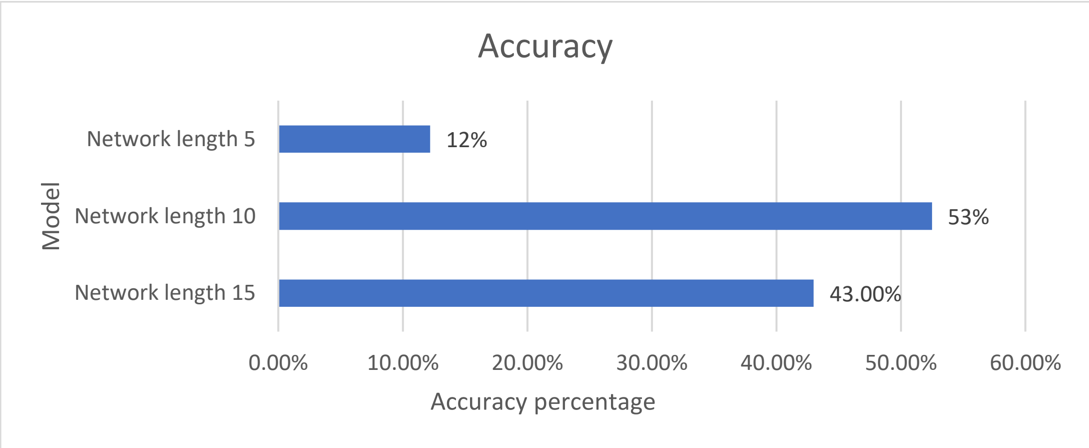

Figure 19. A diagram describing the accuracies of the destination prediction models with lengths 5, 10 and 15.

As the model with length 10 that corresponds to 0.4 seconds in time has the best accuracy rate, further analysis has been done using this model. The following Figure 20 illustrates the accuracy rates for all roundabouts in the dataset.

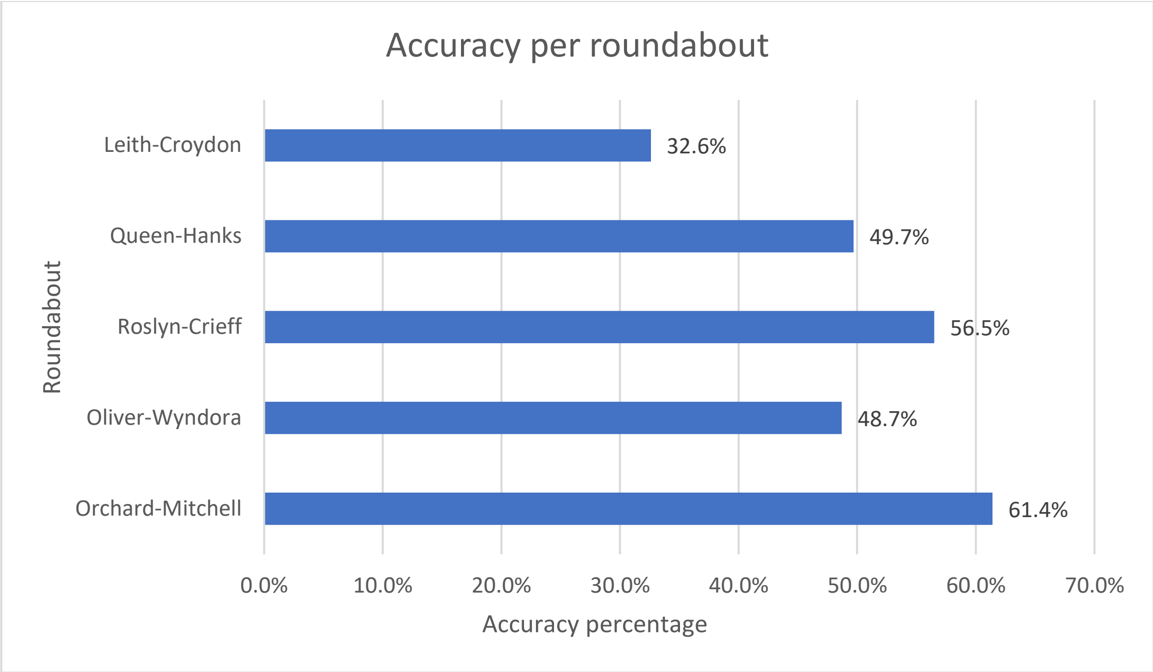

Figure 20. A diagram describing the accuracy of the model with length 10 per every roundabout.

An important metric to be considered when dealing with time-series data is to measure, how long does it take to converge to the ground-truth value. The convergence times for all three models of different network lengths have been compared in Figure 21.

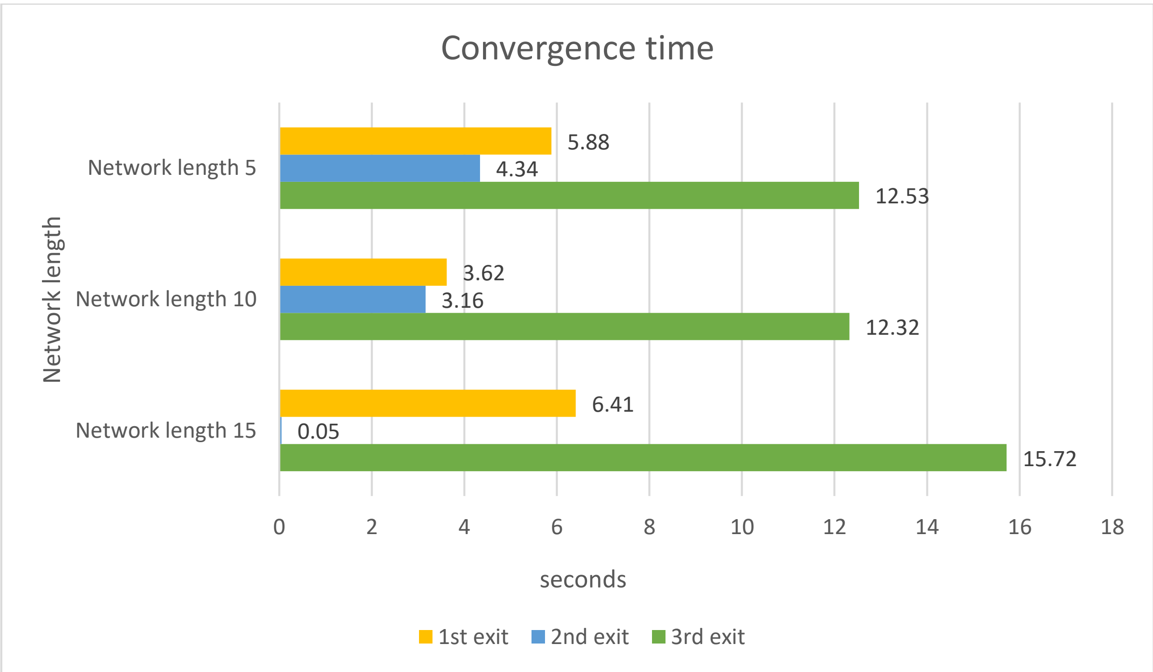

Figure 21. Comparison of models with network length 5, 10 and 15 regarding the convergence time to ground-truth value. For each model, the convergence time has been calculated for the first, second and third exit.

In addition, the tracks in the testing dataset have been grouped by their origin and destination to first, second and third exits. The results are illustrated on Figure 21.

7.2 Action prediction

Another model trained in this research was a model based on Support Vector Machines and the goal of the model was to predict the future action of a vehicle: wait, slow down or speed up.

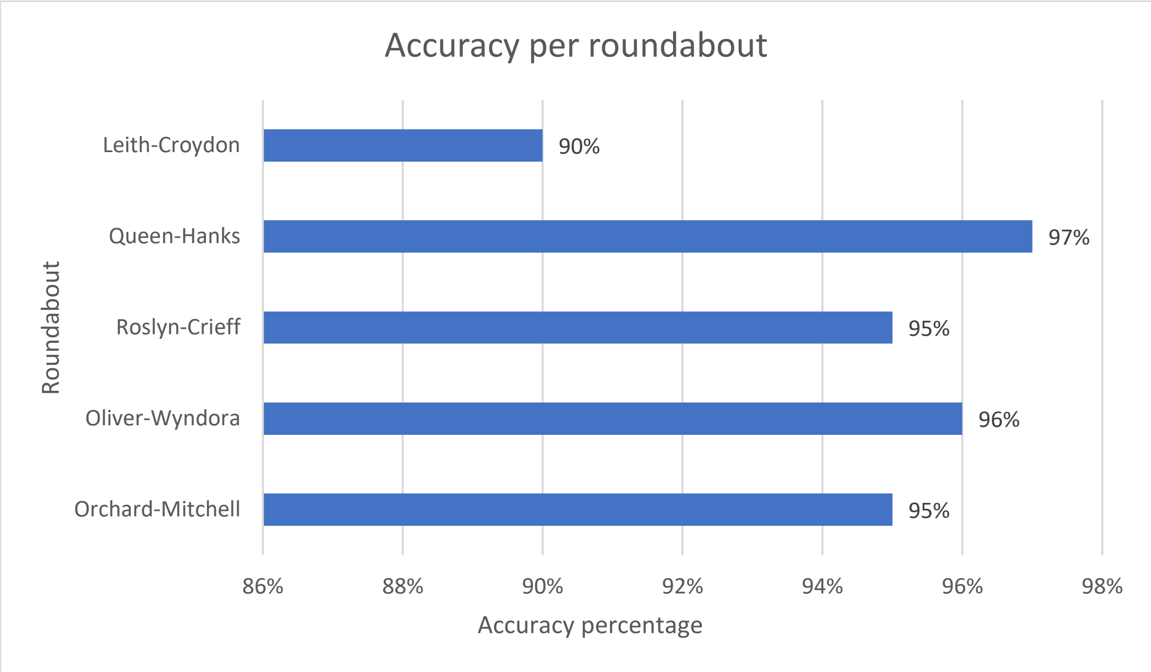

Figure 22. SVM model results for every roundabout.

The model's overall accuracy was 84% and the accuracies per each destination is described in Figure 22.

8 Discussion

The results of the destination prediction model show that the best network length for this task is 10, which corresponds to 0.4 seconds in time. The accuracy for the model with length 15 (0.6 seconds) is very close, however the model with network length 5, corresponding to 0.2 seconds in time, performed the worst. The reason might be that with very little historical data, the model may not have enough information to make an accurate decision. Taking into consideration that the trajectories are not normalized to start from the same exit, resulting in 8 possible exits, predicting the destination for a vehicle is a difficult task even for an experienced human driver. That is why the best model with accuracy of 53% is a very good result.

When comparing the accuracies for each roundabout, it can be seen that the Leith-Croydon roundabout has the lowest accuracy. |The reason might be that this is the roundabout that has the most data in the Five Roundabouts dataset. Before training the classes were balanced so they would have equal amount of data, however the accuracy might drop as the data variety increases. It can also be said that although Roslyn-Crieff roundabout had the least data, the model is able to generalize well since the accuracy for this roundabout is also high.

The comparison of convergence times for the first, second and third exits show that the fastest convergence time is for the second exit, followed by the first one. The reason behind this result may be that the vehicle reaches the first exit fastest and thus, there is too little data to model this behavior. The third exit is the furthest one and the longest convergence time was expected. Overall, the model with network length 10 performs the best also in convergence time, except for the instance of predicting the second exit, where the model with network length 15 performs the best.

The model predicting the most appropriate manoeuvre velocity-wise had excellent results. The overall accuracy was 84% and the accuracies for every roundabout were almost equally good. The lowest accuracy was again for the Leith-Croydon roundabout as this roundabout had the most diverse data and it only had three exits – north, south, and east. Based on these results we conclude that using data from SV can help the autonomous vehicle to choose an appropriate course of action when approaching the roundabout.

9 Future work

In this research we found that both methods work well to suggest a course of action for vehicles on roundabouts. The model used to predict the destination of a vehicle performed relatively well, achieving over 50% in accuracy out of eight possible solutions. The best results can be achieved in less than 0.4 seconds, which is a very short time and thus can be used in naturalistic traffic. However, better results could be achieved if there was more data available. In addition to RNN-based models, the same problem could be solved using SVMs [36], Random Forest [37] or other algorithms that were not covered in this research.

By knowing the most probable destination of a vehicle traversing the roundabout, it is possible to perform further predictions about its future actions. For example, considering the TV's position and the vehicle's predicted destination, it is possible to reach a conclusion whether the vehicle traversing the roundabout has passed the TV or not. In the research we have suggested considering the rightmost quarter of a left-handed traffic roundabout to be the "danger zone". If a vehicle is in this area, then the TV must slow down. In case the destination of the vehicle is the same as the TV's origin, the TV is suggested to wait.

The second method for predicting the future action of a vehicle included using the SVs' data as input to the model. This practice is not very common in research related to autonomous driving, although it is very important for accurately predicting the action of a vehicle. Due to the interdependency of vehicles in traffic, one vehicle might affect the action another vehicle should take. Communication between drivers is especially important in unsignalized intersection, which is why this method has been used in this research. The great results suggest that using this method is very effective and that the SVs' data should be considered if it is available. We suggest further studying the effect of SVs in the context of predicting the future action of a vehicle.

For full integration into autonomous vehicles, there would need to be a differently trained network for each intersection type, to be used as a library. Roundabouts with similar features can be generalized if the dataset consists of data collected from multiple roundabouts of the same type. This research has a focus on the Australian roundabouts, that have a tangential shape opposed to the European radial ones. Since the shape of a roundabout has an impact on the vehicles' speed, the same models might not be applicable to roundabouts of both kind. Furthermore, the traffic rules and drivers' habits in different regions of the world might also affect the driving style even if the roundabouts themselves have similar characteristic. Thus, we suggest testing the models presented in this research on data that is collected from different countries to validate whether the shape of the intersection and driving styles affect the result.

10 Conclusion

With the rapid growth of the automotive industry the need for improved behavior prediction systems have become one of the main challenges in this industry. The ability to predict the intention of drivers at an intersection is critical for the safe driving of an autonomous vehicle, and useful for the ADAS systems. Improvements in hardware have enabled to use more computational power to make predictions about traffic and improved sensors have enabled access and to process more data. Besides following traffic rules, safety and smoothness of motion are also important when designing the intelligent system for autonomous vehicles.

There are numerous factors that contribute to the driving style at roundabouts. In this research, we focused on the Australian roundabouts that have tangential shape that encourages faster travel, and it has good visibility. The dataset we used was collected from five different roundabouts [4] so that any model derived from this data is generalizable.

We proposed two methods with the aim of helping an autonomous vehicle choose an appropriate course of action when approaching the roundabout. Comparing three destination prediction models with different network lengths suggested that good results can be achieved already in 0.4 seconds. The performance of the model predicting the future action in terms of velocity was very good, reaching over 80% in accuracy. It can be concluded that using data from surrounding vehicles may help with modeling the drivers' behaviors.

Further analysis can be done by using the vehicle's destination prediction. This information can be used to determine, whether some vehicles have already passed a target vehicle to avoid collision. We suggest validating the model on European roundabouts since the shape of this roundabout type is very different from the Australian one and the existing model may not be able to generalize well. For full integration into autonomous vehicles, there may arise a need to train new models based on data collected from different types of roundabouts.

11 References

[1] "The Big Problem With Self-Driving Cars Is People," *IEEE Spectrum*, Jul. 27, 2017. https://spectrum.ieee.org/the-big-problem-with-selfdriving-cars-is-people (accessed May 11, 2022).

[2] "Roundabouts | WSDOT." https://wsdot.wa.gov/travel/traffic-safety-methods/roundabouts (accessed May 14, 2022).

[3] N. Muhammad and B. Åstrand, "Predicting Agent Behaviour and State for Applications in a Roundabout-Scenario Autonomous Driving," *Sensors*, vol. 19, no. 19, Art. no. 19, Jan. 2019, doi: 10.3390/s19194279.

[4] A. Zyner, S. Worrall, and E. M. Nebot, "ACFR Five Roundabouts Dataset: Naturalistic Driving at Unsignalized Intersections," *IEEE Intell. Transp. Syst. Mag.*, vol. 11, no. 4, pp. 8–18, 2019, doi: 10.1109/MITS.2019.2907676.

[5] A. Zyner, S. Worrall, and E. Nebot, "A Recurrent Neural Network Solution for Predicting Driver Intention at Unsignalized Intersections," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 1759–1764, Jul. 2018, doi: 10.1109/LRA.2018.2805314.

[6] A. Zyner, S. Worrall, and E. Nebot, "Naturalistic Driver Intention and Path Prediction using Recurrent Neural Networks," *ArXiv180709995 Cs*, Jul. 2018, Accessed: Apr. 02, 2022. [Online]. Available: http://arxiv.org/abs/1807.09995

[7] N. Muhammad and B. Åstrand, "Intention Estimation Using Set of Reference Trajectories as Behaviour Model," *Sensors*, vol. 18, no. 12, Art. no. 12, Dec. 2018, doi: 10.3390/s18124423.

[8] S. Lefèvre, D. Vasquez, and C. Laugier, "A survey on motion prediction and risk assessment for intelligent vehicles," *ROBOMECH J.*, vol. 1, no. 1, p. 1, Jul. 2014, doi: 10.1186/s40648-014-0001-z.

[9] S. Mozaffari, O. Y. Al-Jarrah, M. Dianati, P. Jennings, and A. Mouzakitis, "Deep Learning-based Vehicle Behaviour Prediction For Autonomous Driving Applications: A Review," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 1, pp. 33–47, Jan. 2022, doi: 10.1109/TITS.2020.3012034.

[10] A. Zyner, S. Worrall, J. Ward, and E. Nebot, "Long short term memory for driver intent prediction," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2017, pp. 1484–1489. doi: 10.1109/IVS.2017.7995919.

[11] L. Xin, P. Wang, C.-Y. Chan, J. Chen, S. E. Li, and B. Cheng, "Intention-aware Long Horizon Trajectory Prediction of Surrounding Vehicles using Dual LSTM Networks," *ArXiv190602815 Cs Stat*, Jun. 2019, Accessed: Apr. 27, 2022. [Online]. Available: http://arxiv.org/abs/1906.02815

[12] S. H. Park, B. Kim, C. M. Kang, C. C. Chung, and J. W. Choi, "Sequence-to-Sequence Prediction of Vehicle Trajectory via LSTM Encoder-Decoder Architecture," *ArXiv180206338 Cs*, Oct. 2018, Accessed: Apr. 27, 2022. [Online]. Available: http://arxiv.org/abs/1802.06338

[13] M. Aramrattana, T. Larsson, J. Jansson, and C. Englund, "Dimensions of cooperative driving, its and automation," in *2015 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2015, pp. 144–149. doi: 10.1109/IVS.2015.7225677.

[14] J. A. Michon, "A Critical View of Driver Behavior Models: What Do We Know, What Should We Do?," in *Human Behavior and Traffic Safety*, L. Evans and R. C. Schwing, Eds. Boston, MA: Springer US, 1985, pp. 485–524. doi: 10.1007/978-1-4613-2173-6_19.

[15] E. Ericsson, "Variability in urban driving patterns," *Transp. Res. Part Transp. Environ.*, vol. 5, no. 5, pp. 337–354, Sep. 2000, doi: 10.1016/S1361-9209(00)00003-1.

[16] O. Taubman-Ben-Ari, M. Mikulincer, and O. Gillath, "The multidimensional driving style inventory—scale construct and validation," *Accid. Anal. Prev.*, vol. 36, no. 3, pp. 323–332, May 2004, doi: 10.1016/S0001-4575(03)00010-1.

[17] "Human Factors in Transportation Accidents - Exponent." https://www.exponent.com/services/practices/engineering/human-factors/capabilities/human-factors-in-transportation-accidents (accessed May 15, 2022).

[18] C. Marina Martinez, M. Heucke, F.-Y. Wang, B. Gao, and D. Cao, "Driving Style Recognition for Intelligent Vehicle Control and Advanced Driver Assistance: A Survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 3, pp. 666–676, Mar. 2018, doi: 10.1109/TITS.2017.2706978.

[19] C. Marina Martinez, M. Heucke, F.-Y. Wang, B. Gao, and D. Cao, "Driving Style Recognition for Intelligent Vehicle Control and Advanced Driver Assistance: A Survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 3, pp. 666–676, Mar. 2018, doi: 10.1109/TITS.2017.2706978.

[20] Z. Constantinescu, C. Marinoiu, and M. Vladoiu, "Driving Style Analysis Using Data Mining Techniques," *Int. J. Comput. Commun. Control IJCCC*, vol. V, pp. 654–663, Dec. 2010, doi: 10.15837/ijccc.2010.5.2221.

[21] "What are Neural Networks?," Aug. 03, 2021. https://www.ibm.com/cloud/learn/neural-networks (accessed Mar. 20, 2022).

[22] "Backpropagation | Brilliant Math & Science Wiki." https://brilliant.org/wiki/backpropagation/ (accessed Mar. 23, 2022).

[23] "What are Recurrent Neural Networks?," Apr. 07, 2021. https://www.ibm.com/cloud/learn/recurrent-neural-networks (accessed Mar. 23, 2022).

[24] "Recurrent Neural Network | Brilliant Math & Science Wiki." https://brilliant.org/wiki/recurrent-neural-network/ (accessed Mar. 26, 2022).

[25] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[26] F. Gers, N. Schraudolph, and J. Schmidhuber, "Learning Precise Timing with LSTM Recurrent Networks," *J. Mach. Learn. Res.*, vol. 3, pp. 115–143, Jan. 2002, doi: 10.1162/153244303768966139.

[27] "9.2. Long Short-Term Memory (LSTM) — Dive into Deep Learning 0.17.5 documentation." https://d2l.ai/chapter_recurrent-modern/lstm.html (accessed Mar. 26, 2022).

[28] W. S. Noble, "What is a support vector machine?," *Nat. Biotechnol.*, vol. 24, no. 12, Art. no. 12, Dec. 2006, doi: 10.1038/nbt1206-1565.

[29] "LiDAR sensor technology for autonomous driving - Ibeo," *LiDAR sensor technology for autonomous driving - Ibeo*. https://www.ibeo-as.com/en (accessed Mar. 07, 2022).

[30] A. Wilke, J. Lieswyn, and C. Munro, *Assessment of the effectiveness of on-road bicycle lanes at roundabouts in Australia and New Zealand*. 2014. Accessed: Mar. 10, 2022. [Online]. Available: https://trid.trb.org/view/1312393

[31] N. Halpern-Wight, M. Konstantinou, A. G. Charalambides, and A. Reinders, "Training and Testing of a Single-Layer LSTM Network for Near-Future Solar Forecasting," *Appl. Sci.*, vol. 10, no. 17, Art. no. 17, Jan. 2020, doi: 10.3390/app10175873.

[32] A. Rana and K. K. Kim, "ECG Heartbeat Classification Using a Single Layer LSTM Model," in *2019 International SoC Design Conference (ISOCC)*, Oct. 2019, pp. 267–268. doi: 10.1109/ISOCC47750.2019.9027740.

[33] M. D. Zeiler, "ADADELTA: An Adaptive Learning Rate Method," *ArXiv12125701 Cs*, Dec. 2012, Accessed: May 01, 2022. [Online]. Available: http://arxiv.org/abs/1212.5701

[34] M. Abadi *et al.*, “TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems,” *ArXiv160304467 Cs*, Mar. 2016, Accessed: May 01, 2022. [Online]. Available: http://arxiv.org/abs/1603.04467

[35] “3.3. Metrics and scoring: quantifying the quality of predictions,” *scikit-learn*. https://scikit-learn/stable/modules/model_evaluation.html (accessed May 16, 2022).

[36] L. J. Cao and F. E. H. Tay, “Support vector machine with adaptive parameters in financial time series forecasting,” *IEEE Trans. Neural Netw.*, vol. 14, no. 6, pp. 1506–1518, Nov. 2003, doi: 10.1109/TNN.2003.820556.

[37] P. J. Moore, T. J. Lyons, J. Gallacher, and for the A. D. N. Initiative, “Random forest prediction of Alzheimer’s disease using pairwise selection from time series data,” *PLOS ONE*, vol. 14, no. 2, p. e0211558, Feb. 2019, doi: 10.1371/journal.pone.0211558.

I. License

Non-exclusive licence to reproduce the thesis and make the thesis public

I, Heidi Korp

1. grant the University of Tartu a free permit (non-exclusive licence) to

reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright, my thesis

Agent Behavior Modeling in Roundabout Traffic

supervised by Naveed Muhammad, Phd and Yar Muhammad Phd

2. I grant the University of Tartu a permit to make the thesis specified in point 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 4.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in points 1 and 2.

4. I confirm that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Heidi Korp
16/05/2022